Data Exfiltration by Hotjar Revisited

Alexandra Slezáková and Libor Polčák[1] [a]
[1] *Brno University of Technnology, Faculty of Information Technology, Božetěchova 2, 612 66 Brno, Czech Republic*
{polcak}@fit.vut.cz

Keywords: Web privacy, Session Replay

Abstract: Recently, researchers studied session replay scripts that record interaction of users with websites. This position paper revisits the findings of the original research with the aim of validation of the original results and describing the changes after the publication of the original research. This paper focuses on Hotjar. The default policy to gather inputs changed; the recording script gathers only information from explicitly allowed input elements. Nevertheless, content reflecting users' behavior outside of input HTML elements is recorded. Moreover, we detected improvement in handling TLS. Not only does the web page operators interact with Hotjar through encrypted connections but Hotjar scripts do not work on sites not protected by TLS.

1 INTRODUCTION

Web site operators want to monitor the interaction of the visitors of the web sites, for example to detect problems. Session recording tools allow detail monitoring of user behavior (Filip and Čegan, 2019; Acar et al., 2020). Consequently, session recording scripts are widespread even thought users are typically not aware of the detailed monitoring of their behavior (Acar et al., 2020). Session recording scripts record users' mouse movements, clicks, keystrokes and much more. The collected data is sent to servers of the recording party, which provide aggregated statistics in the form of heat maps or entire recordings. Recordings often contain sensitive user data, such as medical conditions, credit card information and other data that can subsequently be misused (Acar et al., 2020).

(Acar et al., 2020) analyzed the information collected during session recording and revealed password, credit card, and health data leaks. Nevertheless, (Acar et al., 2020) observed that by pointing the flaws during their research, the companies operating data exfiltration services improve the data collection practices. Hotjar is one of the most popular replay service companies in the market. Even so, (Acar et al., 2020) does not list any change in the data collection of Hotjar.

This position paper revisits the findings of (Acar et al., 2020). Firstly, Hotjar changed the default policy of recording the content of all HTML input elements with the possibility to opt out. Currently, Hotjar gathers data from inputs with `data-hj-allow` attribute only. Nevertheless, we show that user's data can still leak, for example when such content modifies the displayed page like during online store check out. In addition, we show that Hotjar respects do not track settings signalled by the browser. Moreover, Hotjar operates through encrypted HTTPS connections and refuses to work on HTTP sites.

The most important contribution of this research is the finding that pointing out flaws in data collection makes sense. Even so, as we highlight the unsolved problem in collecting personal data reflected outside of the input elements, we hope that data collecting companies like Hotjar would address the problem in the future.

This paper is organised as follows. Section 2 provides the necessary theoretical background on session replays, heat maps and Do Not Track web browser setting. Section 3 focuses on related work. Section 4 describes the methodology used to analyse Hotjar with achieved results in section 5. Section 6 discusses our findings. The paper is concluded in section 7.

2 BACKGROUND

This section overviews the application of session recording and how it can improve web pages. In contrast, the section also highlights anti-tracking possibilities that empowers users to block data exfitration.

[a] https://orcid.org/0000-0001-9177-3073

2.1 Session

A web session is a series of requests and responses between a web server. Several techniques exist for maintaining a session state, but sessions are often bound to a cookie. Cookies consist of data stored in the browser that is sent with every request to the server and can be modified with each response. Some applications store an identifier in the cookies that allows the session state to be retrieved by the server (Bortz et al., 2011).

2.2 Session Replay Script

The session replay script records the user's interaction with the website or application and sends it to the server, where it is processed and converted to a format that can be replayed. The principle can be summarised as follows: the user's device is identified by the generated tracking ID, which is stored in the browser. Thanks to this, monitoring user behaviours over a long period is possible (Filip and Čegan, 2019).

Session recordings are usually used by third-party companies whose primary goal is to help understand user behaviours better and thus improve the user experience. This requires gathering data such as whole page source and text, mouse movements and clicks, and key presses, usually without such user's knowledge. Such data collection practice has been particularly problematic in cases involving sensitive data unless the application developer has manually redacted the website or application to ensure the user's privacy (Acar et al., 2020). Gathered data can be analyzed to derive heat maps, recordings of mouse movements, or the success of filling forms.

Session replay scripts can be installed easily. For instance, Hotjar provides the instructions depicted in Figure 1. This code inserts another script into the website. The page environment is initialised to set up the data gathering.

2.3 Heat Map

A heat map is a way to visualise and measure users' interaction with the website. It uses colour intensity to demonstrate the clicks. The brightness of a heat map reflects how popular a particular website section is; therefore, they are essential in determining what works and does not. Without displaying the data numerically, a heat map provides quick and simple-to-understand visuals (Kaur and Singh, 2015), facilitating the analysis and better understanding of how the user interacts with a website. Some specific heat maps used for website analysis to determine a user's behaviour are described in the following sections.

2.3.1 Click Maps

Click heat maps visualize areas where the user clicks the most, as shown in Figure 2. They are based on click coordinates, target element, device resolution and web page content. This information may determine which area of an element is most appropriate for clicking or which element seems clickable (Filip and Čegan, 2019).

2.3.2 Mouse-tracking Heat Maps

Mouse-tracking heat maps, shown in Figure 3, are similar to click maps, but instead of visualizing areas with the most clicks, it visualizes areas where the user hovers the most.

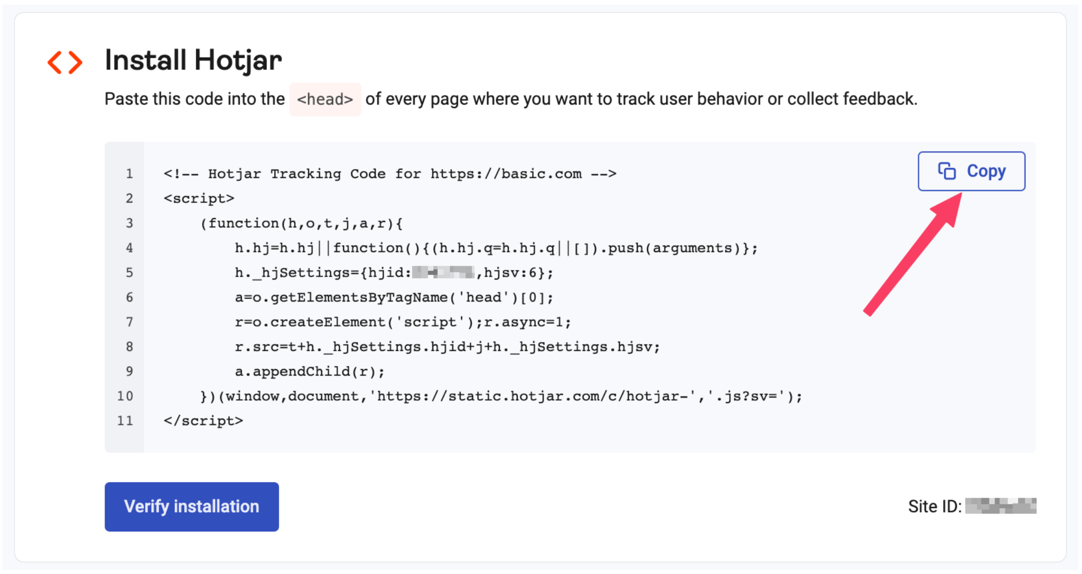
2.3.3 Scroll Maps

Scroll heat maps visually represent the user's scrolling behaviour. It provides statistics about how far a user scrolls down on a website, shown in Figure 4.

2.4 How to Avoid Tracking

Tracker blockers employ lists of URLs or parts of URLs that are considered harmful to user privacy or security. The advantage for the user is that many tools focus on blocking (for example, uBlock Origin, EFF Privacy Badger, Ghostery) and blocklists are usually compatible with several blockers. Browsers like Firefox (Kontaxis and Chew, 2015) and Brave include tracking prevention by default. Previous research shows the by blocking trackers, users improve performance of their browser (Kontaxis and Chew, 2015). The downside of the list-based blockers is that blockers can evade detection by changing the URL of the script so that the rule in the block list no longer matches the URL of the tracker (Merzdovnik et al., 2017).

Do Not Track (DNT) is a web browser setting that allows users to signal opt-out from tracking. DNT was supposed to become a W3C standard (W3C, 2019). The goal of DNT is to enable users to communicate their tracking preferences to each server. A user's web browser sends an HTTP header called DNT. Once a server receives the `DNT: 1` header, it should stop tracking the user. `DNT: 0` means that the user prefers to allow tracking. The standard also allows to access the preference via JavaScript API `navigator.doNotTrack`. Nevertheless, websites do not generally respect the singal (Hils et al., 2021).



Install Hotjar

Paste this code into the `<head>` of every page where you want to track user behavior or collect feedback.

```
<!-- Hotjar Tracking Code for https://basic.com -->
<script>
    (function(h,o,t,j,a,r){
        h.hj=h.hj||function(){(h.hj.q=h.hj.q||[]).push(arguments)};
        h._hjSettings={hjid:[redacted],hjsv:6};
        a=o.getElementsByTagName('head')[0];
        r=o.createElement('script');r.async=1;
        r.src=t+h._hjSettings.hjid+j+h._hjSettings.hjsv;
        a.appendChild(r);
    })(window,document,'https://static.hotjar.com/c/hotjar-','.js?sv=');
</script>
```

Site ID: [redacted]

Figure 1: Hotjar installation script.

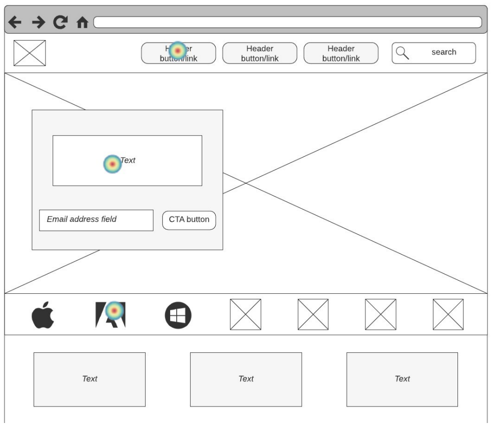


Figure 2: Click map.

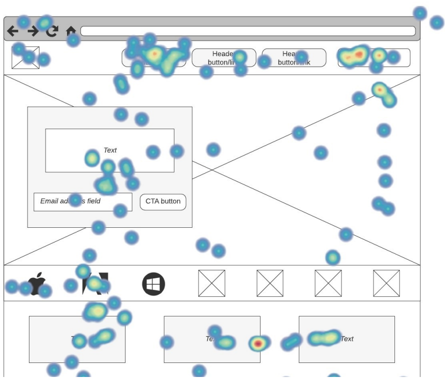


Figure 3: Mouse-tracking heat map.

3 RELATED WORK

(Acar et al., 2020) analyzed six session recording companies in 2018. This study revisits and updates the results of the study of Acar et al. In this position paper, we focus on Hotjar, for two reasons. (1) Acar et al. detected Hotjar on a large number of sites and (2) from our empirical observations we know that Hotjar scripts are present on many commonly visited websites. Additionally, (Acar et al., 2020) does not list any changes in Hotjar collecting practices.

Hotjar collected (1) texts typed into forms before the user submits the form and (2) precise mouse movements. Both without any visual indication to the user(Acar et al., 2020, Section 6.1). At the time of the study, Hotjar collected the text inputs verbatim by default except passwords, credit card numbers, and partially an address. This paper reveals that Hotjar default behavior changed. The content of all inputs is no longer collected by default. (Acar et al., 2020) argued that the opt-out from data collection is not practical. Hotjar likely accepted the arguments and changed its policy, so that it collects only inputs with `data-hj-allow` attribute.

(Acar et al., 2020) also warned that the displayed content on a page can contain user-specific content. For example, online stores typically show the content of the user basket and the filled billing information as a part of the page before during checkout. Our results show that such information still leaks if it is not displayed by input elements.

Moreover, (Acar et al., 2020) argued that pages render different input fields to store passwords. Our experiments use different types of password and explains how passwords can leak to Hotjar.

Even more, (Acar et al., 2020) detected session re-

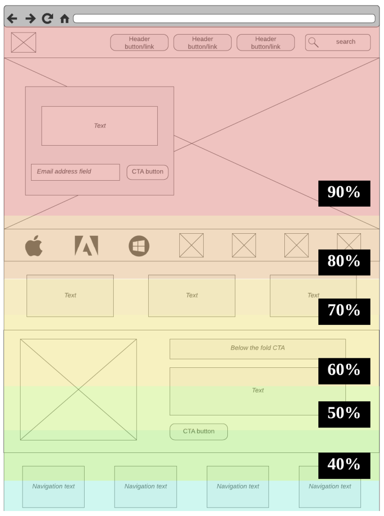


Figure 4: Scroll map.

play companies that transfer content of HTTPS web sites over HTTP, not protected by any encryption. As the original paper does not list Hotjar switching to encrypted access to session replays, we validated the current state.

DNT signal received a little success (Hils et al., 2021). (Acar et al., 2020) did not study if Hotjar and other session collecting companies respect DNT. Nevertheless, Hotjar mentions respect for DNT in its privacy statements. We validated the code and confirmed that Hotjar indeed respects DNT. Even so, we warn that malicious web site operators or attackers can circumvent the DNT detection and force Hotjar to collect data from users signalling their no tracking preferences.

4 METHODOLOGY

Firstly, we studied Hotjar policies, documentation and opened a testing account. We validated that Hotjar respects the Do Not Track signal (Section 5.1). To do so, we read the Hotjar script and located the code responsible for Do Not Track handling. We checked if the interaction with the testing account is encrypted.

Secondly, we analyzed the nature of collected user data with the focus on dynamically generated data by both the user and the page as a result of the users' interaction with the page (Section 5.3).

Finally, we investigate the possibility of the web server operators to modify Hotjar script to increase the amount of collected data.

We created three web applications:

1. A log-in form with a password input and text input for e-mail username. As many log-in forms allow user to show the verbatim password, we included a button that changes the input type of the password input to text.
2. A shipping form where a user needs to fill name, address, e-mail, phone, credit card details and other information typically needed for shipping.
3. A page that changes the content according to the user input. The page contains a text input. As the user types text in, the page offers country names matching the input text.

We created the log-in and shipping form in three varieties:

1. The first application scenario contains a form according to Hotjar's documentation. ID attributes of both input fields were set to the appropriate values, namely, `pass` or `password` for the password input field and `email` for the email input field. According to documentation, Hotjar should not record such input fields and replaces their content with asterisks of arbitrary length. The shipping form also employs specific ID attributes to prevent any potential leakage of sensitive personal information.
2. Even though Hotjar does not record keystrokes by default, the web operator can allow collecting the content of any input element by adding attribute `data-hj-allow`. Consequently, Hotjar records the content that the user inserts into such fields verbatim and does not replace it by asterisks. We analysed the behaviour of `data-hj-allow` attribute by creating another log-in form. However, we did not use any well-known id for the inputs. Instead, we chose IDs randomly and used the attribute `data-hj-allow`. Nevertheless, the type of one of the input was `password`. The form included the same button to show verbatim password as the first form.
3. The two scenarios above use the original Hotjar session replay script. The third form is the same as the one in the first scenario but we modified the Hotjar script. For example, we modified the function that is supposed to mask the username and password. The original function replaces the real text with a random number of asterisks. The modified versions does not transform the original string in any way.

Finally, we tested the default configuration of multiple browsers and validated if the built-in protections prevent pages to record the sessions, see section .

5 RESULTS

We aimed to reproduce the tests mentioned in the study and determine if something has changed in Hotjar to ensure users' privacy.

5.1 Respecting DNT

Unlike many websites that refuse to respect DNT signals (Hils et al., 2021), Hotjar script checks `navigator.doNotTrack` property, see Figure 5. In cases when the property carries the value 1, the recording of a particular session is omitted. Nevertheless, browsers with DNT activated need to download and execute Hotjar script.

```
"1"!==navigator.doNotTrack&&
"1"!==window.doNotTrack&&
"1"!==navigator.msDoNotTrack
```

Figure 5: The script at https://script.hotjar.com/modules.1e98293c16a88afdf1b7.js gathers data only if it does not detect DNT.

The retired DNT standard allows to specify extensions to the `navigator.doNotTrack` property (W3C, 2019, Section 5.2.1). The clause in Figure 5 would not detect such extensions and the preference not to be tracked would not be honored in the presence of extensions (such as `"navigator.doNotTrack == 1xyz"`). Nevertheless, the retired standard warns against using extensions so hopefully, DNT implementers follow the suggestion to not set extensions.

A malicious web site can change the value of `doNotTrack` property to record all sessions, for example, by executing the code in Figure 6 before the inclusion of Hotjar script.

```
Object.defineProperty(navigator,
                     "doNotTrack", {
        get: function () { return "0"; },
        set: function (a) {},
        configurable: false
      });
```

Figure 6: A malicious website can reconfigure browsers to allow tracking by Hotjar.

5.2 TLS Support And Dashboard Data Encryption

Since user data ends in the session recording, recording services must prioritise security. Otherwise, the protection of user data may fail. Hotjar delivered playbacks within an HTTP page[1], even for recordings on HTTPS pages. This allowed a man-in-the-middle to insert a script into the page to extract all the data from the record. Today, Hotjar uses HTTPS.

Moreover, Hotjar script refused to work when deployed on web site without TLS. Hence, a man-in-the-middle adversary cannot misuse Hotjar to record sessions by adding the Hotjar script to HTTP web pages.

5.3 Transferred Data

Hotjar records the position and timestamp of each click or mouse hover and entered data in input fields, including time and timestamp of the input, selector and input field type. While the user's location, operating system and resolution are also recorded, recording a user's location can be changed in Hotjar settings. Then, this information is not transferred.

5.3.1 Usernames and Passwords

The original study (Acar et al., 2020) claims that passwords are excluded from the recordings to prevent password leaks. To validate this statement, we created a sign-in form with two input fields of type email and password and tested different scenarios mentioned in section 4. Here, we provide the test results for each scenario.

Hotjar collecting script masks both inputs (username and password) of the login form respecting Hotjar guidelines. Hotjar displayed asterisks in both input fields. The number of asterisks is different from the actual password length. Hence, the replay does not leak the real length of the password.

The other log-in form contains the password input and both inputs for username and password are decorated with `data-hj-allow` attribute. Even so, Hotjar does not record the content of both input fields. However, once the user interacted with the *show password* feature, our test website changed the type of input fields, causing the user's password to become visible. In this case, Hotjar failed to mask the password field resulting in capturing the full password.

As expected, the modified script that did not replace the input text caused entered data to leak to Hotjar. The usernames, as well as passwords after using the *show password* feature, were then available to the web site operator through Hotjar dashboard.

[1]https://freedom-to-tinker.com/2017/11/15/no-boundaries-exfiltration-of-personal-data-by-session-replay-scripts/, section 4: *"The publisher dashboards for Yandex, Hotjar, and Smartlook all deliver playbacks within an HTTP page, even for recordings which take place on HTTPS pages."*

5.3.2 Shipping details

Firstly, all entered data to the form created according to Hotjar guideliness to prevent any leaks of personal information were replaced with asterisks. Credit card information was replaced with asterisks no matter the data type of the input. Moreover, the phone number was replaced with “1111”.

Since most input fields in the shipping form are of the text type, web operators can allow data collection using `data-hj-allow` attribute. Such change allows the web operator to collect first and last names, phone numbers, company names and credit card information. Data was not masked even when we used `autocomplete` attribute.

The modified session replay script allows adversaries to record all entered data, including the credit card information.

5.3.3 Rendered Website Content

Hotjar collects rendered page content. Unlike user input recording, the collecting script does not suppress the rendered content unless redacted manually, which leaks all displayed content in our tests. In the testing form with one input field, the script did not collect the content of the input field as it lacked the `data-hj-allow` attribute. However, the search results leaked into the recordings. This allows one to guess the content of the input search field.

5.4 Protections Offered by Browsers

Section 2.4 describes a way of how to change the read-only `navigator.doNotTrack` property, which allowed the Hotjar removing the restriction on data transfer. The session was recorded using web browsers such as Google Chrome, Microsoft Edge, Firefox or Opera. However, not every session was recorded. Firefox allowed us to use the tracking protection that blocks content loaded from domains that track users. Even changing the DNT property did not remove the restrictions in this case. Opera also offers tracking protection called Tracker Blocker, which directly blocks a javascript method that adds an additional element to the end of the selected parent element. Then the Hotjar session replay script is not added in the head section of the HTML document.

6 DISCUSSION

Article 29 of the EU Directive 95/46/EC established The Article 29 Working Party (WP29). GDPR transformed WP29 to the European Data Protection Board with increased powers. Both WP29 and EDPB publish guidelines and opinions with the aim of consistent application of data protection law. WP29 published an opinion on the ePrivacy Directive (Directive 2009/136/EC) consent exception. By applying the opinion on session replay scripts, it is clear that users needs to consent to session recording whenever the ePrivacy Directive applies.

(Acar et al., 2020) considered potential violations of laws like GDPR in the EU and HIPAA (healthcare) and FERPA (education) in the United States. To their best knowledge, courts have not decided claims concerning processorship and joint-controllership of session recording companies. In contrast, we believe that the rulings of the Court of Justice of the European Union on the concept of data controllers and processors (Court of Justice of the European Union, 2018; Court of Justice of the European Union, 2019) are applicable to the case of session recording companies. EDPB published guidelines following the judgements (Europan Data Protection Board, 2021). As long as the session recording companies follow the instructions of the web site operators, they can be assumed processors.

As it is the operator that needs to explicitly enable data collection from input elements with the `data-hj-allow` attribute, we believe it is the web operator who controls the data collection and Hotjar is a processor. Nevertheless, Hotjar should allow controlling data collection of other elements and distinguish elements containing user-specific content.

However, distinguishing user-specific non-input elements on the page will likely prove difficult. As explained in Section 2.2 and illustrated in Figure 1, the ease of installation is one of the goals of Hotjar. As the non-input elements typically do not contain user-specific data, the approach of collecting all data by default make sense. The controller can add `data-hj-suppress` attribute to disallow data collection from elements containing user-specific content.

However, as (Acar et al., 2020) highlight in the economics analysis of the failures, web operators lack the budget to hire experts to annotate elements that should not be collected. Selzer et al. (Selzer et al., 2021) studied the costs to comply with GDPR and studied the height of fines and the likelihood of fines. Small and middle-sized companies should only pay a few cents or euros for GDPR legal compliance. Hence, the web operators are not motivated to launch audit on their data collection.

7 CONCLUSION

Session recordings allow web operators to detect problems and optimize web sites with limited budget. However, extensive data collection practices can violate laws like the ePrivacy Directive, HIPAA, and FERPA. (Acar et al., 2020) analyzed data exfiltration of web page visitors several years ago. This position paper focused on Hotjar, a session recording company originally investigated by (Acar et al., 2020). We show that Hotjar changed. As the input elements typically contain user-specific content, they are no longer collected by default. Hotjar employs HTTPS to prevent session data leaks and not allow man-in-the-middle adversaries to inject their scripts. Hotjar respects DNT settings. Nevertheless, we point out several possibilities of adversaries trying to circumvent the protections of Hotjar. User-specific data stored outside input element are still collected by Hotjar by default and we argue that collection of such data will likely continue.

ACKNOWLEDGEMENTS

This work was supported by the Brno University of Technology grant Smart information technology for a resilient society (FIT-S-23-8209).

REFERENCES

Acar, G., Englehardt, S., and Narayanan, A. (2020). No boundaries: data exfiltration by third parties embedded on web pages. *Proceedings on Privacy Enhancing Technologies*, 2020:220–238.

Bortz, A., Barth, A., and Czeskis, A. (2011). Origin cookies: Session integrity for web applications. In *Web 2.0 Security and Privacy (W2SP)*.

Court of Justice of the European Union (2018). Case C-210/16: Unabhängiges Landeszentrum für Datenschutz Schleswig-Holstein v. Wirtschaftsakademie Schleswig-Holstein GmbH. ECLI:EU:C:2018:388.

Court of Justice of the European Union (2019). Case C-40/17: Fashion ID GmbH & Co. KG v. Verbraucherzentrale NRW eV. ECLI:EU:C:2019:629.

Europan Data Protection Board (2021). Guidelines 07/2020 on the concepts of controller and processor in the GDPR. https://edpb.europa.eu/system/files/2021-07/eppb_guidelines_202007_controllerprocessor_final_en.pdf, Version 2.1.

Filip, P. and Čegan, L. (2019). Comparing tools for web-session recording and replaying. In *2019 International Conference on Sustainable Information Engineering and Technology (SIET)*, pages 257–260.

Hils, M., Woods, D. W., , and Böhme, R. (2021). Privacy preference signals: Past, present and future. *Proceedings on Privacy Enhancing Technologies*, 2021:249–269.

Kaur, K. and Singh, H. (2015). Analysis of website using click analytics. *International Journal of Science, Engineering and Computer Technology*, 5(6):185.

Kontaxis, G. and Chew, M. (2015). Tracking protection in firefox for privacy and performance. In *Web 2.0 Security & Privacy Workshop*.

Merzdovnik, G., Huber, M., Buhov, D., Nikiforakis, N., Neuner, S., Schmiedecker, M., and Weippl, E. (2017). Block me if you can: A large-scale study of tracker-blocking tools. In *2017 IEEE European Symposium on Security and Privacy (EuroS P)*, pages 319–333.

Selzer, A., Woods, D., and Böhme, R. (2021). Practitioners' corner: An economic analysis of appropriateness under Article 32 GDPR. *European Data Protection Law Review*, 7(3).

W3C (2019). Tracking preference expression (DNT). The World Wide Web Consortium (W3C), Tracking Protection Working Group, https://www.w3.org/TR/tracking-dnt/.